

Phase Interpolator with Improved Linearity

George Souliotis, Costas Laoudias, Fotis Plessas & Nikolaos Terzopoulos

Circuits, Systems, and Signal Processing

ISSN 0278-081X

Circuits Syst Signal Process
DOI 10.1007/s00034-015-0082-9



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Phase Interpolator with Improved Linearity

George Souliotis¹ · Costas Laoudias¹ ·
Fotis Plessas² · Nikolaos Terzopoulos³

Received: 12 January 2014 / Revised: 16 May 2015 / Accepted: 18 May 2015
© Springer Science+Business Media New York 2015

Abstract An analog phase interpolator with improved step linearity is presented in this paper. The linearity is improved by setting the time constant of the output nodes in suitable value and by employing a fine trimming technique. The performance and the improved linearity have been verified with post-layout simulations using a well-established CMOS 65 nm technology and transistors with standard threshold voltages. The clock frequency is at 2.5 GHz and the core voltage supply at 1.2 V. Its low phase noise makes the circuit suitable for high-speed systems where low jitter performance is required.

Keywords Phase interpolator · Clock and data recovery · SerDes

✉ George Souliotis
gsoul@upatras.gr
Costas Laoudias
laoudiask@upatras.gr
Fotis Plessas
fplessas@inf.uth.gr
Nikolaos Terzopoulos
nterzopoulos@brookes.ac.uk

¹ Department of Physics, University of Patras, Patras, Greece

² Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece

³ Department of Computing and Communication Technologies, Faculty of Technology, Design and Environment, Oxford Brookes University, Wheatley, Oxford, UK

1 Introduction

The phase interpolator (PI) is a critical block in the clock and data recovery (CDR) loop. PI is interposed in SerDes systems, between the PLL and the data samplers, in order to shift the clock phase accordingly in the data sampling window. It receives two clocks of the same frequency with phases ϕ and ψ , respectively, and generates a clock output whose phase Θ is the weighted summation of the two input phases [11]. The ideal PI must generate a number of equally spaced phase steps for a full cycle from 0° to 360° . In fact, the generation of equally spaced steps is a hard process in real PIs, and this is one of the major problems that needs correction for a proper operation. Several digital or analog PI circuits have been proposed in the literature focusing mostly in step linearity [3] with a robust circuit and design flexibility. Some papers propose simple methods to improve linearity, like better control, or feedback from the output, while some others propose more complicated techniques, like multiple step stages [2, 7, 12], but some of them are not suitable for high-speed systems [8]. The specifications for the phase linearity and step size depend on the system architecture, where usually, a number of 32 steps in total is good enough for a typical high-speed de-serializer. The phase step should not be too short to follow the long phase variations within specific clock cycles, but also should not be too long because it may create longer phase steps than the required ones, and then, the clock could be out of the valid data window. A controller is used to drive the steps of the phase interpolator, up or down. This is a fully digital circuit operating as an up/down counter. Also, in most of the high-frequency systems and especially when a spread spectrum clock is used, the phase changes must take place very fast, ideally, in only one bit period in order that the clock will remain in the valid window of the CDR; therefore, both the phase interpolation and its digital controller must be high-speed circuits.

Two generic methods are followed in designing PIs: the analog and the digital. Usually, the analog PIs are employed in high-speed systems with high operating frequency. Some systems use low-frequency PI, but they require a really large number of input phases to feed its input [3]. The large number of phases and the multiple clock distribution make the layout a tough procedure, especially when used in high bit-rate systems. Also, for the digital PIs, highly accurate phases must be generated and distributed from the oscillator. A digital PI proposed in [10] is very simple in design and shows improved linearity but offers small number of steps with large delays, and therefore, it is not suitable for high-speed architectures.

The analog PI is based on the weighted current summation of the two input clocks. The topology is constructed from two arrays of controlled current-mode logic (CML) buffers, as shown in Fig. 1 [1–3, 6, 9, 11, 13]. Usually, two clocks with 90° phase difference feed the input. By selecting the combinations of 0 and 90° phases and their inversions through multiplexers, all the possible phases for a circle of 360 are available, as shown in Fig. 2. The output of the PI is the corrected in phase clock which is a CML signal. Therefore, a CML-to-CMOS converter is required to feed forward all the digital cells. Also, a digital controller controls the operation of the interpolator increasing or decreasing the phase, one step each time.

A flexible, analog-based PI with improved step linearity is proposed in this paper. The improvement is achieved by introducing two techniques; firstly, an integral of the

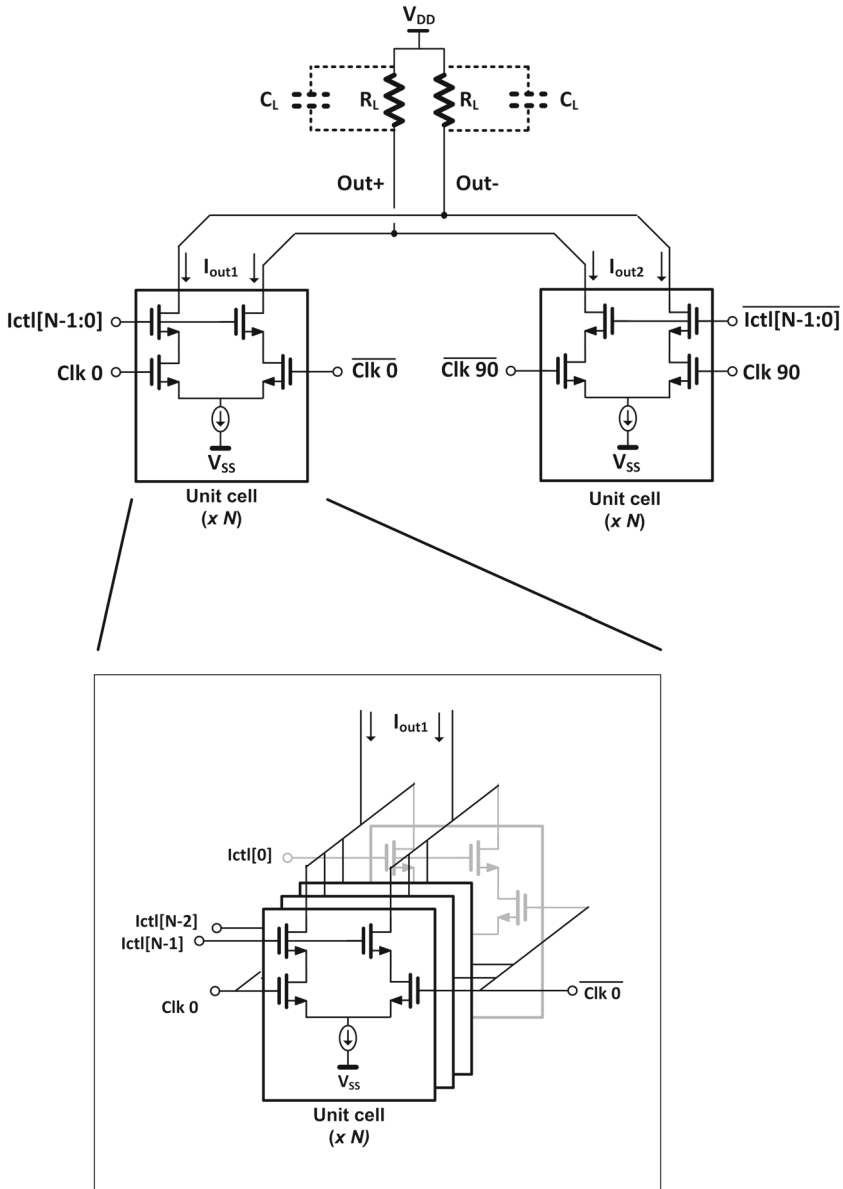


Fig. 1 The analog PI

output signal is produced, and secondly, a fine trimming is carried out by increasing the resolution. The paper is organized as follows: In Sect. 2, the design of the proposed PI is described and the nonlinearity effects are explained. In Sect. 3, the theoretical analysis and some linearity improvements are provided, and in Sect. 3, the simulation results are presented before and after the linearity improvement. Finally, the results regarding the PI performance are summarized.

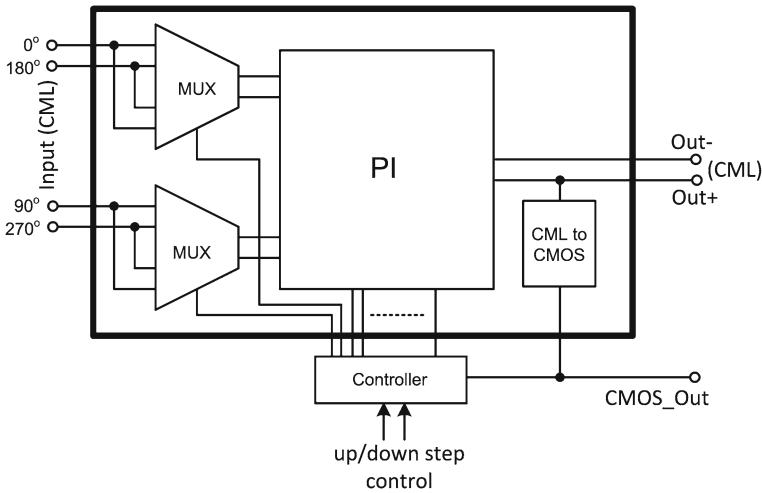


Fig. 2 The topology of the PI

2 The Phase Interpolator

PI receives two clock phases and generates intermediate phases with specific steps. Usually, the input signals, in analog PIs, are of CML type and have a phase difference of 90° in order to produce intermediate phase shifts from 0° to 90° . The full range from 0° to 360° is produced by inverting the inputs, as shown in Fig. 2, giving four quarters of 90° each. The output is also of a CML type and is produced by the weighted current summation of two currents, based on the topology shown in Fig. 1. Two groups of differential pairs are connected on the same output, terminated by resistors R_L . Each group is an array of the same differential pairs with switches connected on their outputs. The differential pair is the PI unit cell and is controlled by I_{ctl} control signal that enables or disables the output current resulting to the total weighted current. In Fig. 1, the total capacitance C_L connected at the output is depicted by dashed lines. This is the total capacitance including all the parasitic effects and has a significant contribution to phase interpolator operation.

The digitally controlled weighted output current is given by,

$$I_{out}(t) = nI_{out1} + (N - n)I_{out2}, \quad n = N:0 \quad (1)$$

where the two currents I_{out1} and I_{out2} have phase difference 90° . N and n are integers, and the intermediate phases generated by the phase interpolator is $N + 1$ resulting to N phase steps. In (1), both currents have the same amplitude I_p .

The weight of currents I_{out1} and I_{out2} is controlled by enabling or disabling the corresponding number of PI unit cells, through the ports I_{ctl} depicted in Fig. 1. A PI controller produces the corresponding signals every time an up/down step is required. Its most simplified version is a thermometer coder which enables or disables only one unit cell each time to increase or decrease the phase step.

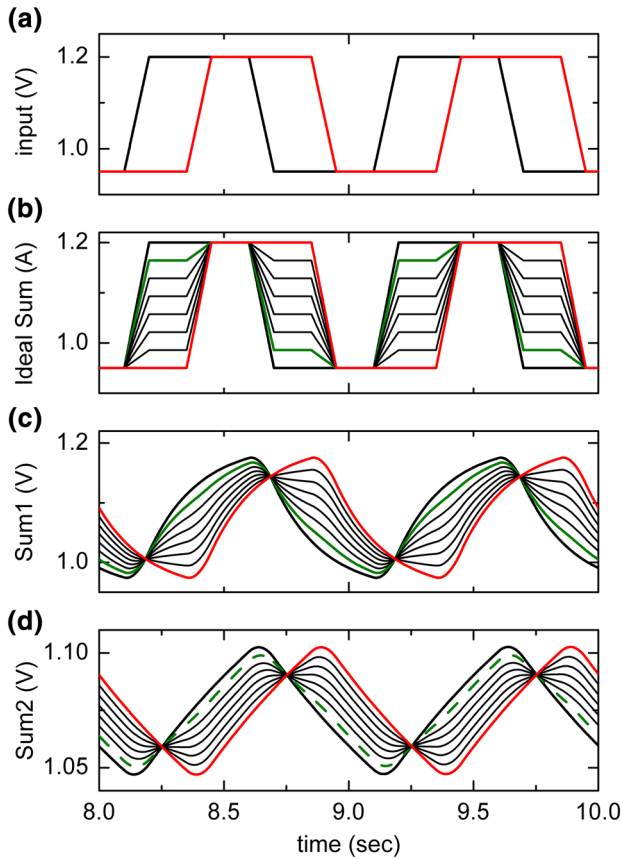


Fig. 3 PI **a** input, **b** ideal output, **c** real output and **d** improved output

A more careful view on (1) shows some potential problems. Assuming that the ideal input signals are shown in Fig. 3a, where the phase 0° signal is depicted in black and the 90° in red color, then according to (1), the resulted ideal current is a stepwise signal consisting of the weighted contribution of each input, as shown in Fig. 3b. The output voltage also takes a similar form according to the relation $V_{\text{out}} = I_{\text{out}} \cdot R_L$. In Fig. 3b, a case of eight phases is shown, with the phase number 0 to correspond to phase 0° and the phase number 7, depicted in red color, to phase 90° . For a better understanding, the phase number 1 is depicted in green color. The stepwise output in its ideal form is not viable in the analog or digital domain, because these steps cannot be processed by a typical following stage, in example a buffer, which requires a normal rising or falling edge to distinguish the low from high state. However, in practice, the ideal output is not expected to be shown in a real high-speed circuit, because of the parasitic capacitances of the output nodes. Instead, an integral of this signal is produced, as shown in Fig. 3c. In that case, the parasitic capacitances act beneficially at the output nodes, creating the output in Fig. 3c, which can be exploited to create phase delays, clearly separated from each other. For example, the phase step 1, indicated with the green line, is now clearly separated from the previous and the next step.

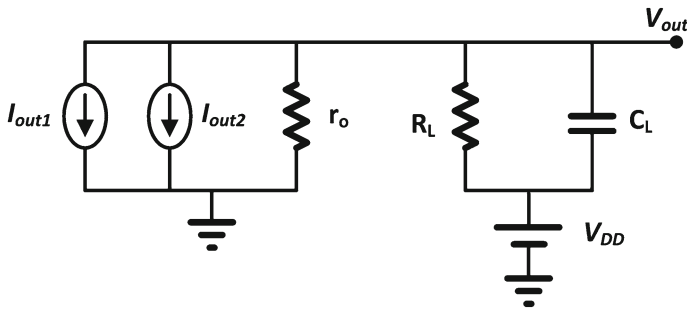


Fig. 4 Equivalent circuit of PI output

The inherent problem at PI's output is that although the final phase steps are available, each step has a different shape, different rise and fall time and does not follow the previous with a uniform way. Therefore, each step does not produce the equal delay.

A simplified analysis for the nonlinearity is provided in [13] which predicts an error on the phase step, due to the exponential form of the output. In [3,4,7], it is reported that the interpolator cells should be asymmetrically weighted, 40% for leading phase and 60% for lagging phase, in order to place the interpolated clock phase in the middle of two adjacent clock phases. However, this asymmetric structure is sensitive to device mismatch, and this increases the risk of a misalignment of the interpolated phase. Thus, interpolator cells with symmetrical structure were finally preferred in [7] to eliminate that risk.

3 Theoretical Analysis and Improvements

The output topology of the PI in Fig. 1 can be represented by the equivalent circuit shown in Fig. 4. For simplicity and without losing the concept, only one of the differential outputs is shown. V_{out} is the output voltage corresponding to the output nodes Out, I_{out1} and I_{out2} are the output currents generated by the output transistors, R_L is the terminating resistor, r_o is the total equivalent output resistance of the transistors in the interpolator unit cells, and C_L is the total capacitance at the output nodes, including all the parasitic capacitances and any other capacitive load charging the output nodes. After a routine analysis, it can be found that the differential equation of the circuit can be given as,

$$R_L C_L \frac{dV_{out}}{dt} = V_{DD} - \left(\frac{R_L}{r_o} + 1 \right) V_{out} - R_L (I_{out1} + I_{out2}) \quad (2)$$

As $R_L \ll r_o$, it is $\frac{R_L}{r_o} + 1 \cong 1$. Then, this first-order linear constant-coefficient ordinary differential equation (LCCODE) can be given as,

$$R_L C_L \frac{dV_{out}}{dt} + V_{out} = V_{DD} - R_L (I_{out1} + I_{out2}) \quad (3)$$

The homogeneous solution can be found by setting the right side to zero. The solution in (3) is given by setting,

$$V_{out}(t) = Ae^{-t/\tau} \tag{4}$$

where A is a constant and $\tau = R_L C_L$ is the time constant. To resolve for the total response, the undetermined coefficient can be a function of time as,

$$V_{out}(t) = A(t) e^{-t/\tau} \tag{5}$$

Substituting (5) into the differential Eq. (3) results in

$$\tau \left[A'(t)e^{-t/\tau} - \frac{1}{\tau} A(t)e^{-t/\tau} \right] + A(t)e^{-t/\tau} = V_{DD} - R_L(I_{out1} + I_{out2}) \tag{6}$$

From (6), it can be easily found that

$$A(t) = \frac{1}{\tau} \int_{x=0}^t [e^{x/\tau} (V_{DD} - R_L(I_{out})) dx] + V_{DD} \tag{7}$$

The last term V_{DD} in (6) is an initial condition assuming that the capacitor C_L is charged on that voltage for $t = 0$, according to Fig. 4.

Finally, substituting (7) into (5) results in the solution of the differential equation,

$$V_{out}(t) = \frac{1}{\tau} e^{-t/\tau} \cdot \int_{x=0}^t [e^{x/\tau} (V_{DD} - R_L(I_{out})) dx] + V_{DD} e^{-t/\tau} \tag{8}$$

Also, I_{out} is the summation of the drain currents of the output transistors shown in Fig. 1.

$$I_{out} = I_{out1} + I_{out2} \tag{9}$$

The output currents in their ideal form can be described from weighted pulses. If N is the total number of phases from 0 phase to 90° phase and n is the specific phase, then I_{out1} and I_{out2} can be given by the following expressions,

$$\left. \begin{aligned} I_{out1} &= \frac{((N-1)-n)}{N-1} I_p \left[u\left(t - \frac{T}{2}\right) - u\left(t - \frac{T}{4}\right) \right], \quad n = 0, 1, 2, \dots, N-1 \\ I_{out2} &= \frac{n}{N-1} I_p \left[u\left(t - \frac{T}{4}\right) - u\left(t - \frac{3T}{4}\right) \right], \quad n = 0, 1, 2, \dots, N-1 \end{aligned} \right\} \tag{10}$$

where I_p is the amplitude of currents I_{out1} and I_{out2} , according to (1).

In frequency domain, the transfer function corresponds to a first-order lowpass filter,

$$H(s) = \frac{1}{\tau s + 1} \tag{11}$$

where τ is the time constant. If τ takes a low value, for example 0.2, then the output is of the form as shown in Fig. 3c, which is similar to the expected real circuit.

A way for improving the shape of the output signal is to make the edges even smoother by taking the integral of that output. This can improve the linearity of rise and fall edges, resulting in an improved step linearity. Enlarging the time constant τ , for example by five times, then the output takes the form of Fig. 3d, which shows an improved shape with less edges comparing with that of Fig. 3c. The change of the time

constant τ is easily realized, in a real circuit, by simply adding an extra capacitor on the output nodes. The drawback here is that, although a high capacitor improves the linearity, the output swing is reduced significantly, because the output node operates obeying the lowpass transfer function, as defined in (11). Therefore, there is a trade-off between the time constant and voltage swing. The exact value of the additional capacitor is not critical; however, a suitable value can be found by the time constant τ defined at the output node. If τ_o corresponds to the operating frequency ω_o of the phase interpolator, then $\tau_o = 1/\omega_o$ and the suitable value for the total capacitance C_L is found by setting $\tau = 2\tau_o$, which is a good compromise between linearity and output swing,

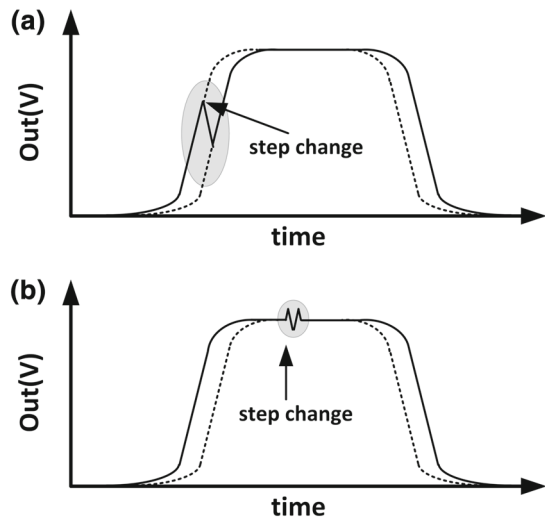
$$C_L = \frac{2}{R_L \omega_o} \quad (12)$$

In (12), C_L is the total value, including the additional capacitor and the parasitic capacitive load. Therefore, the final calculation of the additional capacitor can be found by taking into account the parasitic elements which can be found accurately after the layout extraction. Nevertheless, the additional capacitor is the dominant term in this procedure.

However, even after that method, the linearity is still not greatly improved. A relatively easy procedure to improve the linearity even more is to create a higher step resolution and to keep the best eight steps among them for phases from 0° to 90° . Designing a PI with the number of steps multiplied by 4 (32 steps) for 0° to 90° , by equally divided currents which produce again nonlinear steps, is a relatively straight forward procedure. The selection of the 8 steps among the 32 steps, not necessarily in equally increasing sequence, can give steps with approximated linear relation. The final 8 steps are selected with an order suitable to produce equally shifted steps.

The exact, increased resolution is decided by the step error requirement. Each unit cell shown in Fig. 1 splits k times into sub-unit cells. In this design, k was set equal to 4 to comply with the step error specifications, resulting to 32 steps in total for 0 to 90° . All the transistors of the sub-unit cell have k times smaller width than the corresponding transistors of the unit cell and also are biased by k times smaller current. So, after the split, there are employed totally $(k \times N)$ sub-unit cells meaning that the controller must control $(k \times N)$ devices. Except this controller overhead, the total design is same in terms of layout dimensions and power consumption, because there are k times more cells which, however, are k times smaller in layout and current dissipation. The important point is that the phase interpolator finally dissipates the same current after the cells splitting, keeping the same performance. The controller, however, must be modified because as explained previously, it is programmed to select groups of sub-unit cells resulting to the best N steps among the totally available $k \times N$ steps in the high-resolution topology. Each group contains different number of sub-unit cells depending on the compensation which is required. If no compensation is required for a specific step, then the group contains k sub-unit cells otherwise contains less or greater number than k . Finally, the new controller is again a thermometer coder, but modified in order to enable/disable more than one sub-unit cells at every up/down step coming from a finite state machine (FSM). In the following section, simulation results are provided for the high-resolution phase interpolator which selects the best low-resolution steps.

Fig. 5 **a** Glitch and **b** correction during step change



4 Simulations Results

The analog PI has been designed in a 65 nm CMOS technology, and post-layout simulation results are provided to verify the proposed interpolation method. The supply voltage was 1.2 V, and standard threshold voltage transistors (SVT) were employed. In our system, an 8-step PI was required for the intermediate phases from 0° to 90° at a frequency of 2.5 GHz. The full-cycle phase steps from 0° to 360° is realized by changing the inputs with their opposite signals.

Three issues were to be dealt efficiently in this topology where their combination creates a challenging design: the relatively low supply voltage of 1.2 V in a high-speed circuit, the common mode variations and the glitches during the phase step. The low supply voltage does not allow the cascade connection from rail to rail of a large number of SVT. Therefore, the use of PMOS transistors as active load [11] was not followed in this design. Thus, passive terminating resistors were employed which, however, cannot control effectively the common mode variations. The main advantage of using PMOS transistors is the common mode compensation. Both the common mode and the output amplitude are significantly varied by step to step alteration. The reason is that each step drives different current from the two weighted, 90° phase-shifted outputs, and there is not a uniform current drive from step to step. The solution for the common mode variations is the output to be sensed sufficiently by a suitable next stage buffer.

The timing of the PI controller also is important to avoid glitches of PI output. The controller is clocked by a system clock, and if special care is not taken, it is certain that some of the step switching will occur exactly during the rise or fall edge, as the steps make a full cycle from 0° to 360° . The result, in such a case, is a glitch during the edges [3], as shown in Fig. 5a. It is important to avoid this effect as it gives an uncertain logic value. The glitch can be avoided if the switching takes place always in the middle of the pulse, as shown in Fig. 5b, where there is no side effect in the

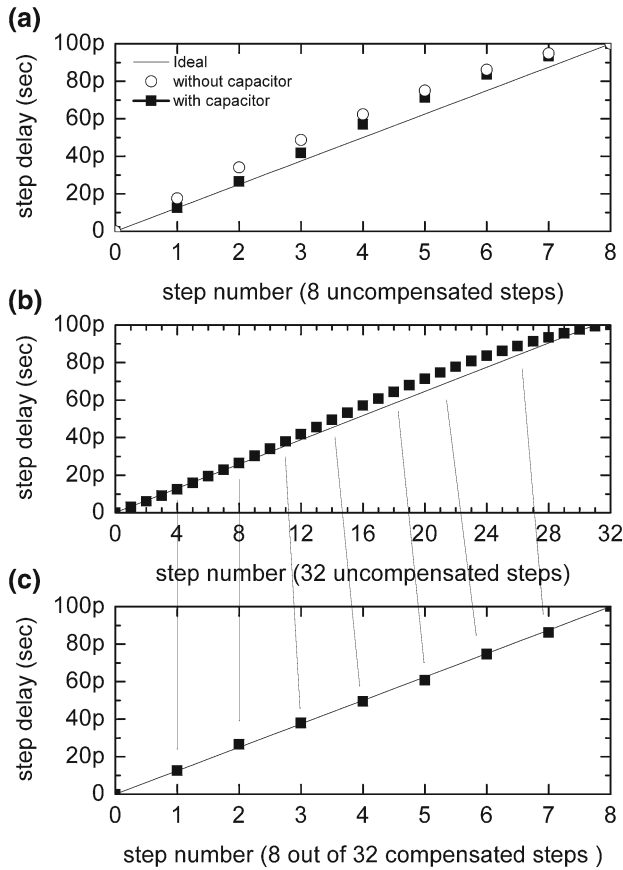


Fig. 6 Step configuration **a** uncompensated 8 steps, **b** uncompensated 32 steps and **c** compensated 8 steps

operation. To ensure this, the output clock from the PI can be used to synchronize the PI control bits, as shown in Fig. 2.

The simulated steps of the PI are shown in Fig. 6a. The initial uncompensated steps are indicated with the circled marks. After adding the extra capacitor, the steps are improved, as shown in Fig. 6a with the square marks. Although there is an improvement, still some steps have a significant distance from their ideal delays. Then, the PI is modified to give extra resolution from 8 to 32 steps for phases from 0° to 90° . These all steps are shown in Fig. 6b. Among them, only 8 are required and the best of them are selected for the final delays, as shown in Fig. 6c. Thus, the linearity is significantly improved, and the final step error is minimized. For example, selecting the bits 0, 4, 8, 11, 14, etc., instead of the bits 0, 4, 8, 12, 16, etc., we take the linear phase shifting as shown in Fig. 6c. The steps without and with the proposed technique, for a full cycle of 360° , is shown in Fig. 7a, b, respectively. The time responses of the 8 steps for delays from 0° to 90° are shown in Fig. 8.

The typical step of the PI is equal to 12.5 ps. From Fig. 9, the maximum phase nonlinearity error of the uncompensated PI with capacitor is 0.72 LSB (9 ps), while of

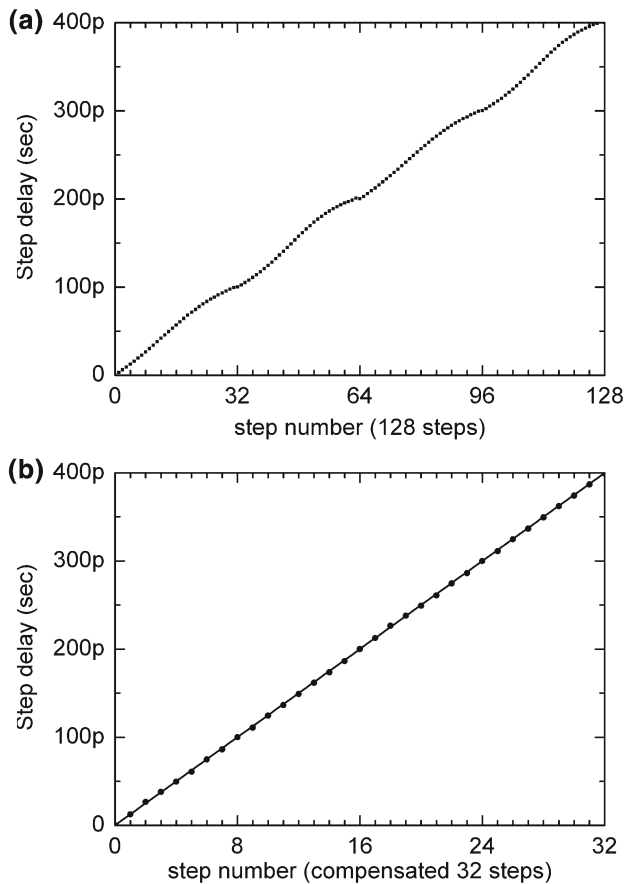


Fig. 7 Step linearity improvement for 360° **a** uncompensated 128 steps, **b** compensated 32 steps

the compensated PI is less than 0.12 LSB (1.5 ps). Monte Carlo simulations performed to estimate the sensitivity on mismatches and device variations. A single typical step was simulated for 100 runs with random mismatches and variations, following the Gaussian distribution in the range of the mean value of +4 sigma. The expected typical step values were 12.5 ps, and are in agreement with the results shown in Fig. 10, where the mean value was 12.56 ps and the standard deviation σ was 0.35 ps.

The phase noise of the phase interpolator must be kept low in order to minimize its contribution to the total noise budget of the system. The phase noise was simulated setting the phase interpolator at a locked, constant phase number for each measurement. Totally, several steps were simulated to find the phase noise level which was found equal to -135 dBc/Hz at 1MHz for typical operation conditions. Corner simulation results, presented in Table 1, show that the phase noise kept low in all cases for slow–slow and fast–fast technology parameters, from -40 to 125°C temperature variations and from 1.08 to 1.32 V voltage supply. The worst case for the phase noise is -131 dBc/Hz at fast–fast 1.08 V voltage supply and 125°C temperature.

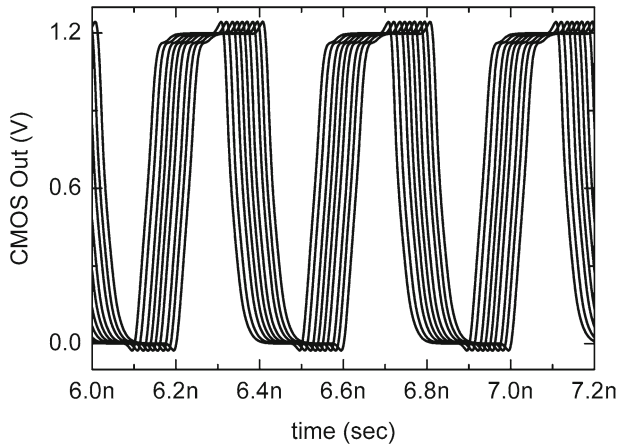


Fig. 8 0° – 90° steps of the PI

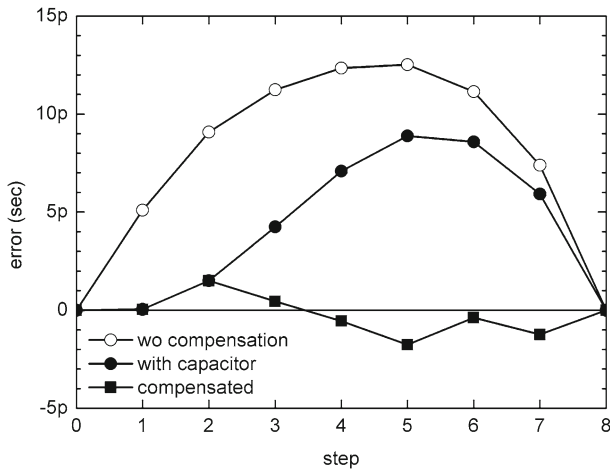


Fig. 9 Step error of phase interpolator

The performance of the PI is summarized in Table 2, and in Fig. 11, the layout is depicted, with some supporting cells. The supporting cells include input CML multiplexers, buffers and CML-to-CMOS conversion cells. Also, the extra capacitors used to improve linearity are shown in this figure. The die area of the core of the PI is $145\ \mu\text{m} \times 35\ \mu\text{m}$, while the total area with the supporting cells, like buffers and CML-to-CMOS converters, is $295\ \mu\text{m} \times 95\ \mu\text{m}$.

A comparison of the performance of the proposed design with other phase interpolators is presented in Table 3. This table includes only papers which give details about the phase interpolator and not refer to it just as a simple operating block in a bigger system. In Table 3, it is presented the typical step, and in the last column, it is presented the error in ps and the relative error comparing with the typical value. The phase interpolator in [3, 11] runs at low speed, and therefore, the control is easy. In

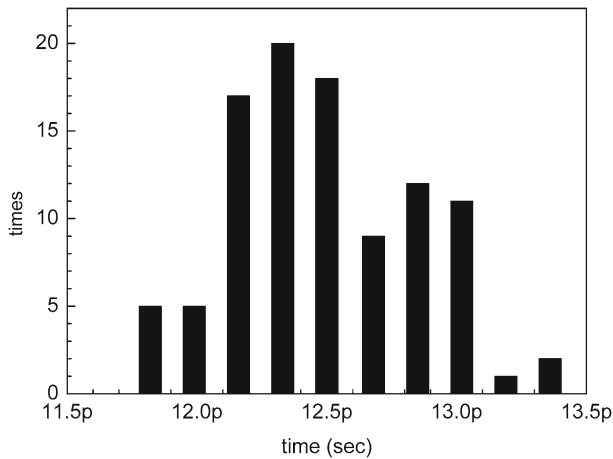


Fig. 10 Monte carlo results of the step variations

Table 1 Phase noise in several corners

Technology variation	Voltage supply (V)	Temp. (°C)	Phase noise @ 1MHz
Typical	1.2	27	-135.7
Fast-fast	1.08	-40	-135.5
Fast-fast	1.08	125	-131
Fast-fast	1.32	-40	-136.4
Fast-fast	1.32	125	-132.5
Slow-slow	1.08	-40	-136.5
Slow-slow	1.08	125	-133.9
Slow-slow	1.32	-40	-137.3
Slow-slow	1.32	125	-135.2

Table 2 Performance of PI

Parameter	Value
VDD (V)	1.2
Tech. (nm)	65
Power (mW)	10
Clock (GHz)	2.5
Typical step (ps)	12.5
Step error of a typical step (ps)	<1
Area ($\mu\text{m} \times \mu\text{m}$)	145 \times 35

[11], several proposed methods improve the phase linearity, which initially was very low. However, this is a first attempt to create phase interpolation by an accurate way. In [3], although a high-resolution interpolator is proposed, no information is given about the error in linearity. High-resolution interpolator is proposed in [7]. Because

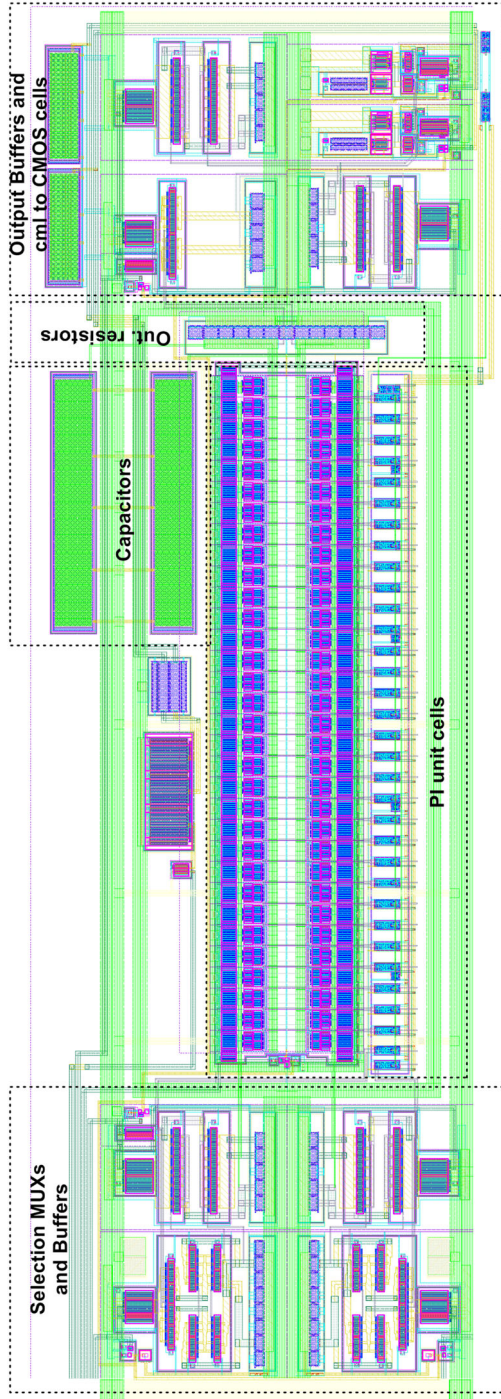


Fig. 11 PI layout

Table 3 Comparison table

References	Tech.	VDD (V)	Power (mW)	Clock (GHz)	Typical step (ps)	Step error (ps), (%)
[1]	0.8 μm	3.3	na	<0.4	22.5	20.25, (90%)
[3]	0.25 μm	2.5	na	0.125	62.5	na
[7]	130 nm	1.2	<10	3.125	3.2	1.2, (37%) ^a
[12]	110 nm	1.2	na	2.5	1.56	1 ps, (64%) ^b
[2]	65 nm	1.2	22.5	5	6.25	na
[13]	65 nm	0.5	na	na	>1000	na
[1]	65 nm	1.2	na	4	20.83	4.51, (21%)
[6]	65 nm	1	na	6	13.88	0.93, (7%)
[9]	130 nm	1.2	na	5	3.125	0.83, (27%)
This work	65 nm	1.2	10	2.5	15.625	na
					12.5	1, (8%) ^a

^a Including mismatch/process variations

^b Worst-case timing equal to 8 ps

of the high resolution, the absolute step error is improved, but the relative error takes values up to 25%. Moreover, high resolution is achieved by feeding the interpolator with clocks that have phase difference 30° , needing extra input clock phases, extra circuits and additional control. Also, in [12], a high-resolution topology is utilized with additional feedback control of the linearity, showing small nonlinearity error taken by spice simulations and worst-case timing variation 8 ps due to 50-mV VDD step. A phase interpolator at 10 GHz is presented in [2], but no information about the step error is provided. In [8], a 0.5V phase interpolator is proposed based on controlling the slew rate of the output, through controllable inverters. However, this operates at low speed, probably due to the low supply voltage and needs multiphase oscillator. A high-frequency phase interpolator which tested at 4 and 6 GHz is referenced in [1]. It exhibits an improved performance at 6 GHz that is significantly reduced at 4 GHz, and therefore, it must be optimized for each frequency clock. The step resolution, however, is only 30° which offers simplicity in terms of design and operation, but on the other hand, this may be too low for many SerDes systems. In [6], a 5 GHz phase interpolator with high resolution and 27% worst step error are reported, and a phase interpolator operating at 1 GHz is given in [9]. Other references are not placed in the table because they operate in low speed, which is out the scope of this work or they use the phase interpolator as a block without giving specific information about its implementation [5]. With regard to Table 3, the proposed phase interpolator shows a good performance for high-speed clock and improved relative linearity. If required, it could be further improved by using a higher resolution step.

5 Conclusions

A design technique for improving the linearity of analog phase interpolators is presented in this paper. The technique focuses on two main actions: One is by setting suitable integration of the output signal and second is by creating higher step resolution. The phase interpolator operating at 2.5 GHz with a 1.2 V voltage supply has been designed using a 65 nm CMOS technology. The phase error is less than 1 ps, and the phase noise is -135 dBc/Hz.

References

1. B. Abiri, A. Shivnaraine, R. Sheikholeslami, H. Tamura, M. Kibune, A 1-to-6Gb/s phase-interpolator-based burst-mode CDR in 65 nm CMOS, in *Proceedings of Solid-State Circuits Conference Digest of Technical Papers*. pp. 154–155 (2011)
2. M. Benyahia, J. B. Moulard, F. Badets, A. Mestassi, T. Finateu, L. Vogt, F. Boissieres, A digitally controlled 5 GHz analog phase interpolator with 10 GHz LC PLL, in *Proceedings of International Conference Design & Technology of Integrated Systems in Nanoscale Era*. pp. 130–135 (2007)
3. H. Chung, D.-K. Jeong, W. Kim, An 128-phase PLL using interpolation technique. *J. Semicond. Technol. Sci.* **3**(4), 181–186 (2003)
4. B.W. Garlepp, K.S. Donnelly, J. Kim, P.S. Chau, J.L. Zerbe, C. Huang, C. V. Tran, C.L. Portmann, D. Stark, Y.-F. Chan, T.H. Lee, M.A. Horowitz, A portable digital DLL architecture for CMOS interface circuits, in *Proceedings of 1998 Symposium on VLSI Circuits. Digest of Technical Papers*. pp. 214–215 (1998)

5. P.K. Hanumolu, G.-Y. Wei, U.-K. Moon, A wide-tracking range clock and data recovery circuit. *IEEE J. Solid State Circuits* **43**(2), 425–429 (2008)
6. S. Hu, C. Jia, K. Huang, C. Zhang, X. Zheng, Z. Wang, A 10 Gbps CDR based on phase interpolator for source synchronous receiver in 65 nm CMOS, in *Proceedings of IEEE International Symposium Circuits Systems*. pp. 309–312 (2012)
7. Y. Jiang, A. Piovaccan, A compact phase interpolator for 3.1256G Serdes application, in *Proceedings of Southwest Symposium Mixed-Signal Design*. pp. 249–252 (2003)
8. S. Kumakil, A.H. Johari, T. Matsubara, I. Hayashi, H. Ishikuro, A 0.5 V 6-bit scalable phase interpolator, in *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems*. pp. 1019–1022 (2010)
9. L.N. Li, W.P. Cai, A phase interpolator CDR with low-voltage CML circuits. *J. Electron. Sci. Technol.* **10**(4), 314–318 (2012)
10. A. Nicholson, J. Jenkins, A. van Schaik, T.J. Hamilton, T. Lehmann, A 1.2 V 2-bit phase interpolator for 65 nm CMOS, in *Proceedings of IEEE International Symposium Circuits Systems*. pp. 2039–2042 (2012)
11. S. Sidiropoulos, M. Horowitz, A semidigital dual delay-locked loop. *IEEE J. Solid State Circuits* **32**(11), 1683–1692 (1997)
12. H. Takauchi, H. Tamura, S. Matsubara, M. Kibune, Y. Doi, T. Chiba, H. Anbutsu, H. Yamaguchi, T. Mori, M. Takatsu, K. Gotoh, T. Sakai, T. Yamamura, A CMOS multichannel 10-Gb/s transceiver. *IEEE J. Solid State Circuits* **38**(12), 2094–2100 (2003)
13. C.-K.K. Yang, *Design of High-Speed Serial Links in CMOS*. Technical report: CSL-TR-98-775. Computer Systems Lab, Department of Electrical Engineering and Computer Science, Stanford University. <http://i.stanford.edu/pub/cstr/reports/csl/tr/98/775/CSL-TR-98-775.pdf> (1998). Accessed 20 May 2015